REMARKS

We have amended the claims to more particularly point out and distinctly claim the invention. The amendments of claim 1 are designed to reorder some of the language to make the claim read better and to make more explicit what was implied by the original wording of that claim.

The Examiner rejected claims 1-5and 17-19 under 35 U.S.C. §103(a) as being unpatentable over Ferris et al. (U.S. 6,253,228) in view of Strong (U.S. 6,167,523). The Examiner admits that Ferris does not specifically teach:

> whether the text input received to the GUI element is valid text input in response to receiving the programmatic event; and
> providing an indication that the text input received to the GUI element is invalid if the text input is determined to be invalid.

So, to supply that which is missing, the Examiner relies on Strong.

The Examiner characterizes Ferris as teaching "a method for automatically validating text input." We believe, however, that the Examiner has read into Ferris teaching that is simply not there. In other words, Ferris is missing more than the Examiner has acknowledged. In addition, we also submit that Strong does not, in fact, supply that which is missing from Ferris.

The Examiner says that Ferris discloses "instantiating a validation manager" and for support he directs our attention to the following passage:

> The name attribute of the WEBOBJECT tag binds a WEBOBJECT HTML template
> entry to the declarations file. For example, the name having a value of "INPUTFIELD"
> (from the WEBOBJECT tag of Table 1) binds its entry to the INPUTFIELD entry in the
> declarations file. A similar approach can be taken for the BUTTON and
> OUTPUTFIELD values associated with the remaining WEBOBJECT tags. The
> INPUTFIELD, BUTTON, and OUTPUTFIELD declarations bind the WEBOBJECT tag
> to instances of the WOApplet class. The WOApplet class permits the specification of
> applet-specific parameters, such as the dimensions of the applet and the location of the
> ".class" file to download to the browser. It also allows you to initialize parameters to be
> downloaded to the applet and to bind an applet's keys to variables and methods in the
> server. (Col. 8, lines 29-59)

But we could find nothing in that passage (or any other passage to which the Examiner has directed our attention) that refers to a validation manager or to validating. Rather, it refers to

7

applets which provide for the specification of certain parameters, e.g. dimensions of the applet and location of the class file to download to the browser, which relate to the input process.

Ferris teaches a method for updating and synchronizing information between a client and a server" (see title). The applets which Ferris describes are meant to be a superior alternative to the prior art approach of using HTML elements such as the HTML FORM element. According to Ferris:

> The FORM element has many limitations that the present invention addresses using individual applications or applets that run on a client. The applets are defined outside of a Web page and can be programmed to produce an unlimited number of input mechanisms to a user (e.g., checkboxes, textboxes, buttons, etc.). [emphasis added] (Col. 4, lines 47-53).

> The present invention provides a method for synchronizing information between a client browser and a server (the client and server can be the same or different computer systems). To avoid the limitations present with the FORM element, the preferred embodiment of the present invention uses individual applets that retrieve user input and an applet controller that manages communication between the applets and the browser and server. (Col. 6, lines 29-33).

In other words, Ferris relates to new input mechanisms which avoid limitations that are associated with the mechanisms that are associated with HTML.

The Examiner also directs our attention to the following paragraph as disclosing "instantiating the validation manager":

> The applets can be written using a programming language such as Java. The advantages of a programming language such as Java include: the ability to write an unlimited number and variety of programs, Java applications will run on almost any supporting platform without the need to recompile the code, Java is widely used in WWW applications, and a Java program may be embedded into a Web page (HTML document). Further, a Java program may execute its action logic on either a client or a server (although applets often execute logic on a client's computer). (Col. 6, lines 33-65)

But again, we could find nothing in that passage that either mentions or even indirectly refers to a validation manager or validating input.

According to claim 1, as amended, at least part of the validation process involves:

> ...in response to receiving the programmatic event at the validation manager, determining whether the received text is valid text input; and providing an indication that the received text is invalid if the received text is determined to be invalid.

8

Ferris simply does not do this. Indeed, a search of the Ferris patent reveals that Ferris never even uses the words "validate" or "validation."

As we noted above, we also disagree that Strong supplies that which is missing. Though Strong does discuss automatic validation of text at the server to which the form is submitted, the process that Strong discloses is not what is recited in the present claims. Thus, even if one were to combine Strong with Ferris, as proposed by the Examiner, the result would not be the claimed invention.

The validation performed by Strong is done on a server that is remote from the client and it is performed in response to receiving form 280 into which input was entered at client PC 200. In Strong's system, entering the text does not trigger any validation or cause the instantiation of a validation manager, as is required by the claims. It is only after the completed form, which has been processed to receive user input, is submitted to the server that validation takes place. When the submitted form is received, server 205 invokes form validation and processing program 255, which checks whether the data includes a first registry key identifier. That first registry key identifier, along with other information, indicates what information that is stored in server registry 270 should be used to process the form and to validate the data.

This procedure that is implemented by Strong's system is missing at least three important features that are required by the claims. First of all, Strong's form does not include "a markup language tag for instantiating a validation manager." Secondly, Strong's form validation and processing program 255 at server 205 is not instantiated "in response to processing the markup language file," rather it is invoked in response to receiving the submitted form at the server (at that point the processing which is referred to in claim 1 would already have taken place at the client). And thirdly, Strong does not send a programmatic event or anything else to form validation and processing program 255 "in response to receiving the text that is input to the GUI element."

In other words, as stated above, even if the features of Strong wee added to the Ferris system, the result would not be the claimed invention.

For the reasons stated above, we believe that the claims are in condition for allowance and therefore ask the Examiner to allow them to issue.

9

Please apply any charges not covered, or any credits, to Deposit Account No. 08-0219.

Respectfully submitted,

Dated: November 20, 2006

Eric L. Prahl
Registration No.: 32,590
Attorney for Applicant(s)

Wilmer Cutler Pickering Hale and Dorr LLP
60 State Street
Boston, Massachusetts 02109
(617) 526-6000 (telephone)
(617) 526-5000 (facsimile)

10